Graph Anonymization of Small World Network Using Particle Swarm Optimization

Sukhendu Jana[1], Soumaya Mandal[2]

[1]Electronics & Communication Engineering Department
Kalyani Government Engineering College
Kalyani, West Bengal, India

[2]Electronics & Communication Engineering Department
Kalyani Government Engineering College
Kalyani, West Bengal, India
[1]jana.sukha93@gmail.com, [2]s.mandal711312@gmail.com

Abstract *–Now a days, the Social networks are the most essential and popular resources of communication. Huge amount of data, along with sensitive information, is transferred, shared and exchanged between the social media users. With the increasing number of users, the chance of data abuse is also increasing. Therefore the protection of data is needed. Several methods are invented for that purpose. One of the method to protect data is through anonymizing it. Data anonymization can be done in different ways. One of the new approaches of graph- anonymization is employing PSO (Particle Swarm Optimization). Particle Swarm Optimization is an evolutionary method which was first introduced in 1995 by Kennedy and Eberhart. This idea is fundamentally generated from a basic social system like, characterization of either bird- flock or from fish-school. Since then research is going on this technique, derivation of new version, development of new applications, theoretical studies of the effects of the various parameters used it this technique and many more. This paper proposed a concept to resolve the problem of graph anonymization issue by using PSO.*

Key Terms – Graph annonymization, Particle swam optimization, Edge annonymization, Vertex annonymization, Security attack;

1. Introduction

There are different kinds of problems related tograph theory. A number of algorithms are there to solve these problems. One of the popular problems is The Graph anonymization Problem. The difficulty is evolving around the arrangement, ordering and a discrete set of values' selection from a bounded set which is accustomed to fulfill a desired goal. The combinatorial optimization problem is a problem for finding an optimum solution from a discrete set of values. Network designing for fleet management, crew scheduling, optimal performance etc. are the practical scenarios in which various combinatorial optimization problems are applied. Though several combinatorial optimization problems are resolved in polynomial time, many of these belongs to the category of NP –hard. By using estimation and heuristic algorithms, as a compromise within the solution quality and computational time. [1] These hard optimization problems are resolved. The effectiveness of the heuristic algorithms totally relies on it's converging ability to the optimal solution and it's computational complexity. In the area of combinatorial optimization a new heuristic algorithms, named as the meta-heuristic algorithms, are developed to demonstrate some promising results. The class of meta-heuristic algorithms includes simulated tabu search, Genetic algorithms, Evolutionary algorithms, Scatter search, Ant Colony Optimization, and Iterated local search. From this group of meta-heuristic algorithms, the PSO is a computationally efficient algorithm to handle non-linear restraints and has a developed solution space. So, the ideas of the standard PSO model has an extension to the discrete combinatorial space and new models are developed to resolve the combinatorial optimization problem.[1]

Social network sites are used daily for entertainment, socialization, advertising and many more. These social network sites are loaded with huge amount of personal information, which can be presented by social graphs (vertices represent persons, and edges represent relationships).Different privacy attacks cause disclosure of data from social graphs. Irrespective of the privacy protection terms, the sensitive relationships represented by edges may still disclose. Therefore, anonymization methods are invented and studied to protect the delicate edges. Three issues are basically checked in the method design: privacy measure, utility loss measure and performance. Operators of online social networks are increasingly sharing potentially sensitive information about users and their relationships with others. Anonymization is typically used to protect privacy, i.e. removing names, addresses, DOB etc. Several online and almost all offline networks restrain to access the information about individuals and their relationships. [2]

2. Graph Annonymization

In recent years different graphs became significant as a tool for representing information. Users of social Networks are represented as nodes of those graphs. To preserve the privacy ofusers when one is publishing information, especially in the case of social graphs, graph anonymization is needed. Therefore, it is essential to implement an anonymization process in the data in order to preserve users' privacy. Anonymization is a kind of information sanitization to protect the privacy of social network users. This process either encrypts or removes personally identifiable information from data sets, so that the owner the data remains anonymous. Data anonymization is such a technology which converts clear plain text data into an unhuman readable and unchanged form which includes some cryptographic hash functions and techniques where the decryption keys are discarded. This process also enables the transfer of information across a boundary like between two departments within an agency or between two agencies, along with the reducing the risk of unintended disclosure, in certain environments in a manner that enables evaluation and analytics post-anonymization.

It is very easy to anonymize a social graph before publishing to hide private information. However, in [3, 4, 5], there has been demonstrated that a trivially anonymized graph does not completely protect privacy against a vertex re-identification attack.. In background there are two classes of knowledge which was used in vertex re-identification attacks i.e. vertex attributes and topological graph features. The labels of vertices can be viewed as tuples in a data table. The attributes of this table may be divided into quasi-identifier (QI) like age, gender, or zip code, and sensitive attribute (SA) like disease or salary. Irrespective of the absence of personal identity in the table, an adversary can re-identify a victim using the quasi-identifier, and discover personal sensitive information. Some statistical models, data mining techniques and computer security attacks can occur information re association attacks and other user's information depending on the user's reputation.

For the privacy preservation of the social graph, two methods are mainly used like, Vertex anonymization method and Edge anonymization method. Tabular data anonymization methods [6] is the one which can be easily applied to anonymize vertex labels to prevent vertex related attacks like vertex re-identification attacks based on different vertex attributes. These methods were developed in the past years based on *k*-anonymity [7] concept. In aspect of graph features there are some vertex re-identification attacks. When it partitions the vertex degree or neighborhood vertices it can occur attacks. Edge anonymization methods can be grouped into two categories depending on whether the published graph is edge-labeled ornot like Edge anonymity of labeled graph and edge anonymity of unlabeled graph.

3. Particle Swarm Optimization

In many cases these combinatorial optimization problems could be resolved in polynomial time but a major parts of these problems exists in the class of NP –hard. With a compromise between computational time and solution quality, these types optimization problems are solved by applying the heuristic algorithms and some approximations. The effectiveness of the heuristic algorithms totally relies on it's converging ability to the optimal solution and it's computational complexity [1]. In the area of combinatorial optimization a new heuristic algorithms, named as the meta-heuristic algorithms, are developed to demonstrate some promising results. In PSO the flow of the algorithm initiates with particles population where the positions of the particles present the possible solutions for the considered problem, and velocities are initiated in the state space randomly.

In every iteration, some specified fitness measurement determines the fitness of position of every particles and the particle's velocity is being updated by tracing for the two best position. A particle's velocity and position are updated as follows.

$$u_{nk} = u_{nk} + A_1b_1(p_{nk} - z_{nk}) + A_2b_2(g_{nk} - z_{nk}) \quad (1)$$

Where, $n = 1,2,\ldots\ldots,N$ and
$k = 1,2,\ldots\ldots,K$

$$z_{nk} = z_{nk} + u_{nk} \quad (2)$$

Where, A1 and A2 are positive constants called accelerating co efficients.
Ns is the total numbers of the particles in the swarm,K is the dimension of problem search space, i.e. number of parameters of the function being optimized b1 and b2 are two independently generated irregular numbers in a range[0,1] .

The other vectors are defined as:
$Z_n = [z_{n1}, z_{n2}, \ldots \ldots, z_{nk}]$ is the position of nth particle;
$u_n = [u_{n1}, u_{n2}, \ldots\ldots, u_{nk}]$ is the velocity of nthparticle;
$p_n = [p_{n1}, p_{n2}, \quad, p_{nk}]$ is the most suited position of the nth particle or pBesti;
$g_n = [g_{n1}, g_{n2}, \quad, g_{nk}]$ is the most suited position considered by the Neighbourhood of the particle i or gBesti.

Eq. (1) determines the new velocity for every particle by using its earlier velocity, the particle's position wherewith the most probable fitness is reached as yet, and the neighbours' most suitable position is achieved. Eq. (2) updates each particle's position in the solution hyperspace. The use of hard bounds on velocity, however, presents some problems.The maximum velocity Vmax is a constraint to control the global exploration ability of a particle swarm. A larger Vmax assists worldwide exploration, although a smaller Vmax promotes local utilization. The idea of inertia weight that was created for better controlling of the exploration and utilization.The involvement of the inertia weight in the PSO algorithm was first stated in a poublications in 1998 [8].

$$u_{nk} = Yu_{nk} + A_1b_1(p_{nk} - z_{nk}) + A_2b_2(g_{nk} - z_{nk}) \quad (3)$$

$$z_{nk} = z_{nk} + u_{nk} \quad (4)$$

Equations (3) and (4) describe the velocity and position update equations with an inertia weight included. The commonly used PSOs are either "global" version or "local" version of PSO [9]. In "global" version, all other particles influence the velocity of a particle. The neighborhood topology of the particle swarm has a significant effect on its ability to find optima that is the optimal pattern of connectivity among individuals which depends on the problem being solved. The population topology is categorized a two types such as static topology and dynamic topology. The first one is generally known as "gbest" topology and the second one is evolving with "lbest" topology [10].

In another paper it was suggested that, as the "lbest" topology seemed better for exploring the state space while "gbest" converged faster. Therefore, it could be very good idea to begin the search with an "lbest" ring lattice and slowly increase the size of the neighborhood, until the population was fully connected by the end of the run. The results on another kind of topology were also reported, where neighbors were defined by proximity in the search space and the number of adjacencies was dynamically increased through the course of the run.

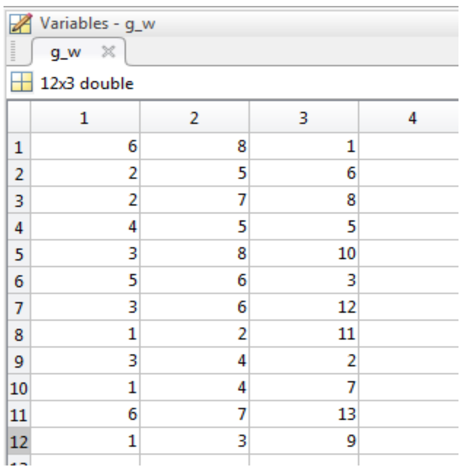
The particles were organised in a hierarchy that is dynamic, with each particle being effected by it's previous success and that of the particle directly above it. Particles with better performance are moved up the hierarchy. The result is that they have more effect on poorer particles. The result was considered as an improved performance on most of the benchmark problems.

The binary particle swarm technique is compared to various kinds of Gas and the criterion allows the generation of arbitrary binary problems with some pre defined characteristics, such that the number of local optimum, dimension, etc. In that study, the binary particle swarm was the only algorithm that found the global optimum on every single trial, regardless of problem features. It also progressed faster than GAs with crossover, mutation, or both, on all problems except the very simplest ones, with low dimension and a small number of local optima. The mutation-only GA was slightly faster in those cases. Many researchers have suggested some modifications to the particle swarm algorithm by allowing it for the operation on binary spaces [11]

4. Implementation Approach

The stated approach is coded in MATLAB (version). There are three different programs, used for the anonymization of simple graph.

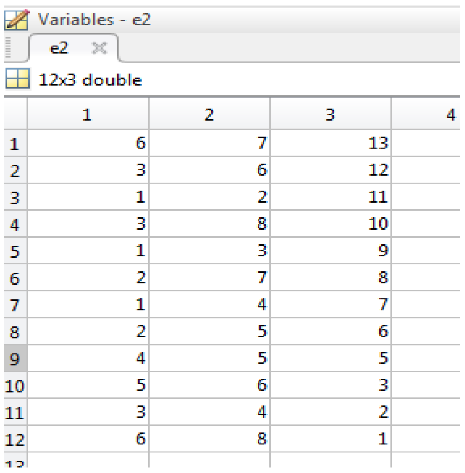
PSO.m: This program is used to optimize the weight of the graphs. It finds out the weight of different edges in a specific order. Input of this program is the data of the graph in matrix form i.e. start node (column 1), end-node (column 2), weights (column 3).


Variables - g_w

g_w

12x3 double

	1	2	3	4
1	6	8	1	
2	2	5	6	
3	2	7	8	
4	4	5	5	
5	3	8	10	
6	5	6	3	
7	3	6	12	
8	1	2	11	
9	3	4	2	
10	1	4	7	
11	6	7	13	
12	1	3	9	


Variables - e2

e2

12x3 double

	1	2	3	4
1	6	7	13	
2	3	6	12	
3	1	2	11	
4	3	8	10	
5	1	3	9	
6	2	7	8	
7	1	4	7	
8	2	5	6	
9	4	5	5	
10	5	6	3	
11	3	4	2	
12	6	8	1	

Figure 1: Simulated Data of Edge Having Max and Min Weight

EDGE_DELETION.m: This program is for the deletion of the edges having less weight. Output of the previous program (PSO.m) is fed as the input of this one.What percent of edges has to be deleted is defined and stored in the variable 'd'.Then the number of rows has to be deleted is calculated and stored in 'n'.After the deletion the output is stored in matrix a2.

GRAPH_PLOT.m: This program plots the graph with the help of the graph-data i.e. start node, end node and weight etc.Start nodes are stored is 's' and end nodes are stored in 'e'.Both the original graph and the anonymized graph after deletion are plotted usingthis.

5. Algorithm

Step 1: BEGIN

Step 2: Initialize the particles' number.
Step 3: Initialize the position and velocity of each particle in the population randomly.

Step 4: Calculate each particle's fitness value .
Step 5: Calculate pBest and gBest for every single particle.
Step 6: loop
Step 7: Update particle's velocity using Eq. (1).

Step 8: Update each particle's position using Eq. (2).

Step 9: Re-calculate the fitness value of every particle.
Step 10: For every single particle update the pBest if its existing fitness value is more than its pBest.

Step 11: Update gBest for each particle i.e. choose the position for the particle with it's best fitness value among all the neighbor particles and assign it as gBest for a specific neighborhood topology.

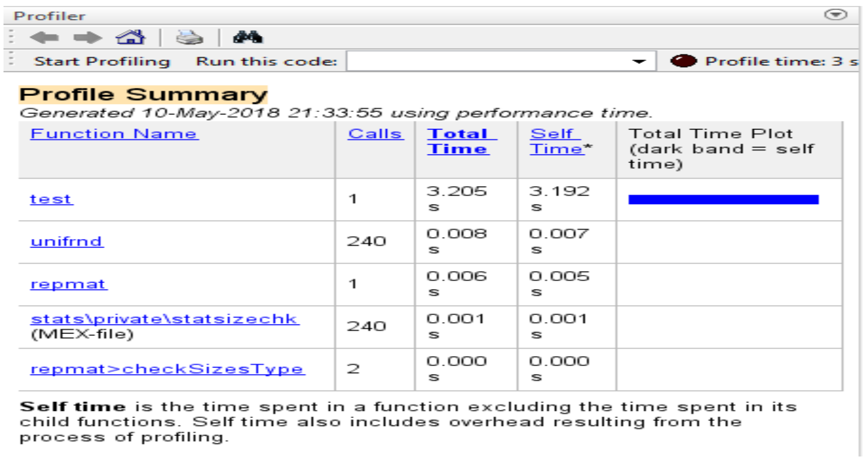
Step 12: If a paradigm is such as an enough fitness or a highest number of repetations have been done then exit loop.

Step 13: end loop.
Step 14: END

Note :- Eq. (1) and Eq. (2), used at Step 6 and Step 7, will change according to the nature of the problem.

6. Results and Discussion

PSO.m optimizes the weights of the input graph within the following time:



Profile Summary
Generated 10-May-2018 21:33:55 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
test	1	3.205 s	3.192 s	▬
unifrnd	240	0.008 s	0.007 s	
repmat	1	0.006 s	0.005 s	
stats\private\statsizechk (MEX-file)	240	0.001 s	0.001 s	
repmat>checkSizesType	2	0.000 s	0.000 s	

Self time is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

Figure 2: Weight Optimization Time of the Graph

Input graph :
g_w= [6,8,1;2,5,6;2,7,8;4,5,5;3,8,10;5,6,3;3,6,12;1,2,11;3,4,2;1,4,7;6,7,13;1,3,9];

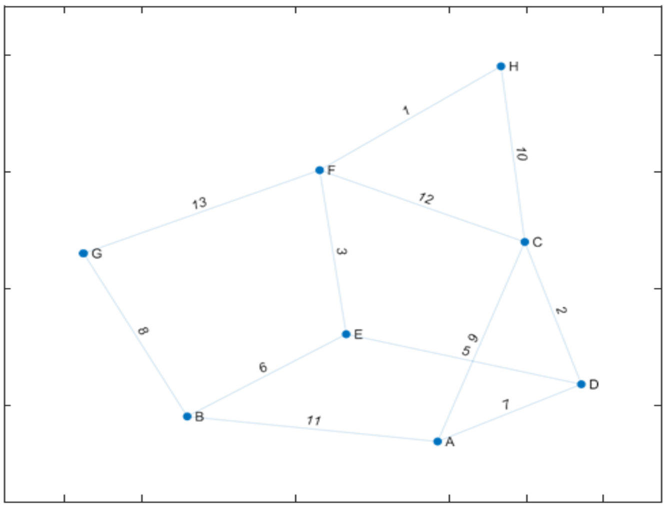


Figure 3: Simulated Input Graph

This graph data is taken as input and then from the output 21% of the total number edges are deleted to get the data of the anonymized graph.

Now, In this case, 3 edges have been deleted and the new graph is as follows,

a2= [6,7,13;3,6,12;1,2,11;3,8,10;1,3,9;2,7,8;1,4,7;2,5,6;4,5,5];

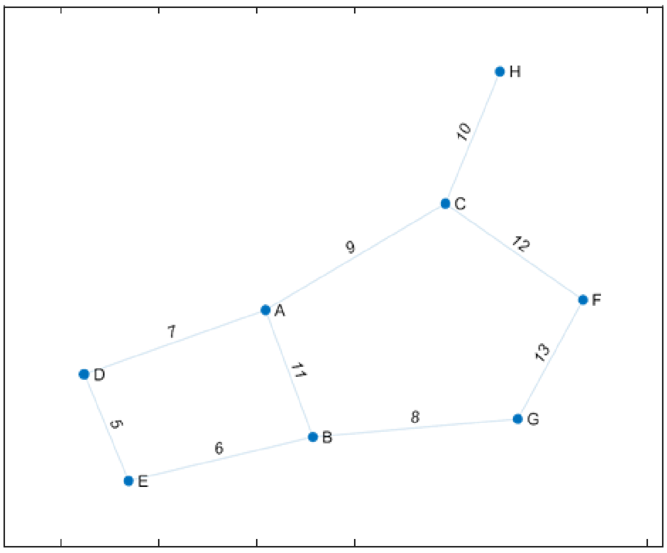


Figure 4: Modified Input Graph

Three edges having weights 1,2 and 3 have been deleted from the input graph.

7. Conclusion

Particle Swarm Optimization method can successfully be used in finding the less important edges. Then the graph is perturbated by deleting those nodes, resulting the required anonymized graph. Particle Swarm Optimization is computationally very cost efficient is respect of speed and the requirements of memory. PSO also needs only some primary mathematical operators. [10]. A better flexible approach is provided to the information system by a number of "cluster" of particles by separating and exploring various region. It is a highly decentralized model which could be run with any number of particles [9]. PSO is simple, easy to implement and requires few lines of codes. PSO provides high-quality solutions within less computational time. The PSO is simple, robust and has low computational costs which make it an ideal method for continuous optimization problems.

References

[1] Festa,P.andResende,M.G.C.(2008).HybridGraspHeuristics.AT&TLabsResearch,FlorhamPark.

[2] LijieZhang,M.Sc.,TheUniversityofTexas,PrivacyPreservationinSocialGraphs,2012.

[3] Lars Backstorm, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In International World Wide Web Conference, 2007.

[4] Michael Hay, Gerome Miklau, David Jensen, Don Towsley and Philipp Weis. Resisting structuralreidentificationinanonymizedsocialnetworks.InInternationalConferenceonVeryLargeDataBases,2008

[5] BinZhouandJianPei.Preservingprivacyinsocialnetworksagainstneighborhoodattacks,IEEE,2008.

[6] C. Aggarwal and P. Yu. A condensation approach to privacy preserving data mining. In InternationalConferenceonExtendingDatabaseTechnology,pages183–199,2004.

[7] P. Samarati and L. Sweeney. Protecting privacy when disclosing information:k-anonymity anditsenforcementthroughgeneralizationandsuppression.InProc.oftheIEEESymposiumonResearchinSecurity and Privac,1998.

[8] Shi, Y., &Eberhart, R. C. (1998b). A modified particle swarm optimizer. In Proceedings of the IEEE international conference on evolutionary computation (pp. 69–73). Piscataway:IEEE.

[9] Russell C. Eberhart and Yuhui Shi, Particle Swarm Optimization: Developments, Applications and Resources, IEEE.

[10] Kennedy, J., &Eberhart, R. C. (1995). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks IV (pp. 1942–1948). Piscataway:IEEE

[11] Riccardo Poli , James Kennedy and Tim Blackwell, Particle swarm optimization: An overview, Springer Science + Business Media, LLC2007.